DebootstrapChroot

This article demonstrates a quick and easy way to create a *chroot* environment on an Ubuntu computer, which is like having a virtual system without the overhead of actual virtualization.

A *chroot* can be used for things like:

- Running a 32-bit Firefox browser or a 32-bit Wine bottle on a 64-bit system.
- Trying an older or newer Ubuntu release without reinstalling the operating system.
- Trying a Debian release or other distribution derived from Debian.
- Cross compiling and building packages for a different platform like Launchpad or Soyuz does it.

Inhaltsverzeichnis

Example Configuration

In this example, we use a current Ubuntu 9.04 Jaunty system (the "host") to create a *chroot* for the older Ubuntu 8.04 Hardy release (the "target"). We are arbitrarily naming the new *chroot* environment **hardy_i386** and putting it in the /srv/chroot directory on the host system.

Step 1: Install packages on the host computer.

First, install `debootstrap`, which is a utility that downloads and unpacks a basic Ubuntu system:

```
$ sudo apt-get install debootstrap
```

Second, install `schroot`, which is a utility that wraps the regular `chroot` program and automatically manages *chroot* environments:

```
$ sudo apt-get install schroot
```

Note: The `debootstrap` utility is usually backwards compatible with older releases, but it may be incompatible with newer releases. For example, the `debootstrap` that is bundled with Jaunty can prepare a Hardy *chroot* like we are doing here, but the `debootstrap` that is bundled with Hardy cannot prepare a Jaunty *chroot*.

If you have any difficultly with a `debootstrap` version mismatch, then visit http://packages.ubuntu.com/ to manually download and install the debootstrap package on the host system from the repository for the target release.

Step 2: Create a configuration file for schroot.

Choose a short name for the *chroot*, we use **hardy_i386** in this example, and create a configuration file for it like this:

```
sudo editor /etc/schroot/chroot.d/hardy_i386.conf
```

Note: In lucid the filename must not contain '.' , it should be lucid_i386_conf.

Put this in the new file:

```
[hardy_i386]
description=Ubuntu 8.04 Hardy for i386
location=/srv/chroot/hardy_i386
#personality=linux32
root-users=bob
run-setup-scripts=true
run-exec-scripts=true
type=directory
users=alice,bob,charlie
```

Note: if you copy this example to your clipboard, be careful to start each line in column 1 before you save the new file! If you forget, the command **schroot -l** will fail with an error, e.g. `E: /etc/schroot/chroot.d /hardy_i386.conf: line 0: Invalid line: “  [hardy_i386]”`.

Note: for lucid use **directory** instead of **location**, e.g. **directory=/srv/chroot/hardy_i386** .

Change these things in the example configuration file to fit your system:

location: This should be a directory that is outside of the /home tree. The latest `schroot` documentation recommends /srv/chroot.

personality: Enable this line if the host system is 64-bit running on an amd64/x64 computer and the chroot is 32-bit for i386. Otherwise, leave it disabled.

users: These are users on the host system that can invoke the `schroot` program and get access to the *chroot* system. Your username on the host system should be here.

root-users: These are users on the host system that can invoke the `schroot` program and get direct access to the chroot system as the root user.

Note: Do not put whitespace around the '=' character, and do not quote strings after the '=' character.

Step 3: Run debootstrap.

This will download and unpack a basic Ubuntu system to the chroot directory, similar to what the host system already has at the real root directory ("/").

```
$ sudo mkdir -p /srv/chroot/hardy_i386
$ sudo debootstrap --variant=buildd --arch=i386 hardy /srv/chroot/hardy_i386 http://archive.ubuntu.com/u
```

This command should work for any distribution that is derived from Debian. Substitute the architecture "i386", the release name "hardy", and the repository address "http://archive.ubuntu.com/ubuntu/" appropriately. For example, do this to get the 64-bit build of Hardy instead of the 32-bit build:

```
$ sudo debootstrap --arch=amd64 hardy /srv/chroot/hardy_amd64/ http://archive.ubuntu.com/ubuntu/
```

Note: Remember to change all instances of **hardy_i386** to **hardy_amd64** in the configuration file and on the command line if you actually do this.

Do something like this to get an upstream Debian release:

```
$ sudo debootstrap --arch=amd64 sid /srv/chroot/sid_amd64/ http://ftp.debian.org/debian/
```

If trouble arises, debootsrap accepts a --verbose flag that may provide further insight.

Step 4: Check the chroot

This command lists configured *chroot*s:tro

```
$ schroot -l
```

If **hardy_i386** appears in the list, then run:

```
$ schroot -c hardy_i386 -u root
```

Note: This should work without using `sudo` to invoke the `schroot` program, and it should result in a root prompt in the *chroot* environment.

Check that the root prompt is in a different system:

```
# lsb_release -a
```

For the Hardy system that we just built, the `lsb_release` command should print:

```
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 8.04
Release:        8.04
Codename:       hardy
```

We're done!

WARNING

For convenience, the default `schroot` configuration rebinds the /home directory on the host system so that it appears in the *chroot* system. This could be unexpected if you are familiar with the older `dchroot` program or the regular `chroot` program because it means that you can accidentally delete or otherwise damage things in /home on the host system.

To change this behavior run:

```
$ sudo editor /etc/schroot/mount-defaults
```

And disable the /home line so that the file reads:

```
# mount.defaults: static file system information for chroots.
# Note that the mount point will be prefixed by the chroot path
# (CHROOT_PATH)
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
/dev/pts        /dev/pts        none    rw,bind         0       0
tmpfs           /dev/shm        tmpfs   defaults        0       0
#/home           /home           none    rw,bind         0       0
/tmp            /tmp            none    rw,bind         0       0
```

The mount.defaults file is the /etc/fstab for *chroot* environments.

Hints

Install the ubuntu-minimal package in a new *chroot* after you create it:

```
$ schroot -c hardy_i386 -u root
# apt-get install ubuntu-minimal
```

If you get locale warnings in the *chroot* like "**Locale not supported by C library.**" or "**perl: warning: Setting locale failed.**" , then try one or more of these commands:

```
$ sudo dpkg-reconfigure locales

$ sudo apt-get install language-pack-en

$ locale-gen en_US.UTF-8
```

If your preferred language is not English, then change "-en" and "en_US" appropriately.

As of Lucid, `schroot` has changed in these ways:

- The file should be named: `/etc/schroot/chroot.d/hardy-i386`
- The keywords in the file have changed and some have been deprecated. Additionally, keywords have to start at the beginning of the line. The file should read:

```
[hardy-i386]
description=Ubuntu 8.04 Hardy for i386
directory=/srv/chroot/hardy-i386
#personality=linux32
root-users=bob
type=directory
users=alice,bob,charlie
```

As of Maverick `schroot` has further changed in these ways:

- The configuration file should be stored in `/etc/schroot/`

TLDR

There's a much simplier way to get a basic chroot environment from an ISO image; if the text above seems TLDR, try this.

First of all, install Ubuntu Customization Kit:

```
$ sudo apt-get install uck
```

Then set the directory in which you want to create the chroot environment:

```
$ export BASEDIR=/path/to/chroot/directory/
```

Unpack the ISO image (this may take quite some time):

```
$ sudo uck-remaster-unpack-iso /path/to/your/image.iso "$BASEDIR" && sudo uck-remaster-unpack-rootfs "$
```

You're done! Now, to enter the chroot environment, just execute

```
$ sudo uck-remaster-chroot-rootfs /path/to/chroot/directory/
```

every time you wish to enter the chroot console. To leave it, type "exit".

To be able to run X applications, e.g. gedit, run

```
# HOME=/root
```

in the chroot environment.

CategoryDevelopment

DebootstrapChroot (zuletzt geändert am 2012-07-03 18:07:00 durch pargue @ 158.52.254.239[158.52.254.239]:pargue)